A tool for automating the computationally complete symbolic attacker (Extended Abstract)

Hubert Comon-Lundh
LSV
CNRS & ENS Cachan

Véronique Cortier
LORIA

Guillaume Scerri
LSV
CNRS & ENS Cachan

The design of automated security proofs is a topic extensively studied for over 20 years. One problem that was raised about 12 years ago is the validity (or the scope) of such proofs. Symbolic models are quite far from the implementation. In contrast, modern cryptography typically considers more powerful attackers. This includes of course some computations that are not explicitly specified. This issue has been first addressed by M. Abadi and P. Rogaway [1], followed by many authors. The idea is to prove that the symbolic formal model is *sound* with respect to the more concrete computational model: if there is no attack in the symbolic model, then there is no attack in the computational model. There are several such soundness proofs, for various primitives and in various contexts (see e.g. [10], [2], [9] to cite only a few). However, all these results require heavy proofs and assume strong hypotheses, some of which are not quite realistic. Typical examples of unrealistic assumptions include: a key cycle is never created, or the attacker does use the key generation algorithm to build his own keys.

These difficulties lead to try to prove the security protocols directly in the computational model. For instance CRYPTOVERIF [6] or EASYCRYPT [5] are designed in this spirit. The proofs have however to account for probability distributions computations, attacker's time computation, and are relatively difficult, often requiring user interactions. We study here an alternative approach presented in [4] which consists in specifying formally what the attacker *cannot do*. Each axiom in such a specification can be a consequence of an assumption on the primitives, which yields the soundness of the model by construction.

Intuitively, checking for cryptographic security in this model amounts to checking the satisfiability of a finite set of first order formulas. In [8] we provided an (efficient) decision procedure for a fragment of first order logic large enough to model reasonable security properties and computational assumptions.

Following these ideas, we present a tool that automates this procedure, together with a set of axioms allowing to prove (and find attacks on) most protocols involving encryption and signatures. As often, our implementation slightly differs from the theoretical algorithm. First, we did not implement the full strategy, losing the polynomial complexity. Second, we have extended our procedure to cope with more axioms such as functionality and reflexivity, needed in our examples. One of the main advantages of our tool is that it allows to find implicit implementation hypotheses. For example, a new attack has been discovered on Needham-Schroeder-Lowe (NSL) in [3] if a nonce can be confused with a pair. Such an attack could not be detected in other symbolic models. We have applied our tool on several protocols from the literature. In particular, we have easily rediscovered the attack on NSL. We have also discovered a new attack on the Andrew secure RPC protocol, if the encryption scheme is not secure when the key is obtained by projecting a nonce. IND-CCA security does not provide any guarantee for this. As a result we conclude that implementations of Andrew's secure RPC should make sure that the second projection of a nonce fails with overwhelming probability. This attack is similar to the one mentioned in [7] but is slightly more general in the sense that it works for more implementations. For example, our attack still holds when nonces are keys. We hope to find more similar implicit hypotheses in the next few months using our tool.

Acknowledgements. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no 258865, project ProSecure.

REFERENCES

[1] M Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. Cryptology*, 15(2):103–127, 2002.

[2] M. Backes and B. Pfitzmann. Symmetric encryption in a simulatable Dolev-Yao style cryptographic library. In *17th IEEE Computer Science Foundations Workshop (CSFW'04)*, pages 204–218, 2004.

[3] G. Bana, P. Adão, and H. Sakurada. Computationally complete symbolic attacker in action. In *32nd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 18 of *LIPIcs*, pages 546–560, 2012.

[4] G. Bana and H. Comon-Lundh. Towards unconditional soundness: Computationally complete symbolic attacker. In *1st Conference on Principles of Security and Trust (POST'12)*, volume 7215 of *LNCS*, pages 189–208, 2012.

[5] G. Barthe, B. Grégoire, S. Heraud, and S. Zanella Béguelin. Computer-aided security proofs for the working cryptographer. In *Advances in Cryptology (CRYPTO'11)*, LNCS. Springer, 2011.

[6] B. Blanchet. A computationally sound mechanized prover for security protocols. In *IEEE Symposium on Security and Privacy (S&P'06)*, 2006.

[7] C. Boyd and A. Mathuria. *Protocols for authentication and key establishment*. Springer, 2003. Page 81.

[8] H. Comon-Lundh, V. Cortier, and G. Scerri. Tractable inference systems: an extension with a deducibility predicate. In *24th Conference on Automated Deduction (CADE'13)*, LNAI. Springer, 2013.

[9] A. Datta, A. Derek, J.C.Mitchell, and B.Warinschi. Computationally sound compositional logic for key exchange protocols. In *19th IEEE Computer Security Foundations Workshop (CSF'06)*, 2006.

[10] D. Micciancio and B. Warinschi. Soundness of formal encryption in the presence of active adversaries. In *Theory of Cryptography Conference (TCC 2004)*, volume 2951 of *LNCS*, pages 133–151, 2004.